CreateFile-01

Always use CREATE_NEW

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-20

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 7090 bytes

Attack Category	• File Manipulation
Vulnerability Category	• Access Control • Privilege escalation problem
Software Context	• File Creation
Location	• winbase.h
Description	The CreateFile function creates or opens a file, file stream, directory, physical disk, volume, console buffer, tape drive, communications resource, mailslot, or named pipe. The function returns a handle that can be used to access an object. Do not use CREATE_ALWAYS or OPEN_ALWAYS as dwCreationDisposition for secure files. Do not pass the CREATE_ALWAYS or OPEN_ALWAYS flags as the dwCreationDisposition. If the file exists, the function overwrites the file, and it does not reset the security descriptor (SD) specified by the SECURITY_ATTRIBUTES structure. This means that a new file could be created with inappropriate security settings. More generally: CREATE_NEW always sets the SECURITY_ATTRIBUTES. CREATE_ALWAYS and OPEN_ALWAYS may or may not set the SECURITY_ATTRIBUTES. OPEN_EXISTING and TRUNCATE_EXISTING never set the SECURITY_ATTRIBUTES.
APIs	**FunctionName** / **Comments** CreateFile CreateFileA CreateFileW

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

<table>
<tr><td>Method of Attack</td><td>An attacker could pre-create a file with the attacker's own permissions. When createfile() is called on this file, it is overwritten as a new file but still has the SECURITY_ATTRIBUTES set by the attacker.

What you ask for, in essence, is not what you get. This can allow attackers access to otherwise secure data.</td></tr>
<tr><td>Exception Criteria</td><td></td></tr>
<tr><td>Solutions</td><td>
<table>
<tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr>
<tr><td>When using createfile() to create a new file.</td><td>Do not use CREATE_ALWAYS,OPEN_ALWAYS, OPEN_EXISTING or TRUNCATE_EXISTING flags as dwCreationDisposition for secure files. These flags are not guaranteed to set the SECURITY_ATTRIBUTES for the new file. Use the CREATE_NEW flag instead. It does set the SECURITY_ATTRIBUTES for the new file.</td><td>Effective</td></tr>
<tr><td>If CREATE_ALWAYS or OPEN_ALWAYS must be used.</td><td>Check the GetLastError() value for ERROR_ALREADY_EXISTS and reset the SECURITY_ATTRIBUTES</td><td>Effective</td></tr>
</table>
</td></tr>
<tr><td>Signature Details</td><td>HANDLE CreateFile(LPCTSTR lpFileName, DWORD dwDesiredAccess, DWORD dwShareMode, LPSECURITY_ATTRIBUTES lpSecurityAttributes, DWORD dwCreationDisposition, DWORD dwFlagsAndAttributes, HANDLE hTemplateFile);</td></tr>
<tr><td>Examples of Incorrect Code</td><td><pre>HANDLE hFile
= ::CreateFile("myfile",
GENERIC_READ | GENERIC_WRITE, 0,
NULL, CREATE_ALWAYS, 0, NULL);

//file handle
HANDLE hFile;

//something to contain the number
of bytes read</pre></td></tr>
</table>

	DWORD dwNumWritten;
	
	//a Boolean test variable, to test for success of reads
	BOOL bTest;
	
	//a buffer… can actually be of any type
	DWORD dwBuffer[256];
	
	// Opening a new file for writing:
	
	hFile = CreateFile("D:\\mypath\
	\\myfile.dat",
	GENERIC_WRITE,
	FILE_SHARE_WRITE,
	NULL,
	// This is the issue ---> DWORD dwCreationDisposition,
	CREATE_ALWAYS,
	FILE_ATTRIBUTE_NORMAL,
	NULL);
	/* if there is a problem opening the file, a call to GetLastError will tell you what it is */

Examples of Corrected Code	// Open an existing file for reading
	
	//file handle
	HANDLE hFile;
	
	//something to contain the number of bytes read
	DWORD dwNumRead;
	
	//a Boolean test variable, to test for success of reads
	BOOL bTest;
	
	//a buffer… can actually be of any type
	DWORD dwBuffer[256];
	
	hFile = CreateFile("D:\\mypath\
	\\myfile.dat",
	GENERIC_READ,
	FILE_SHARE_READ,
	NULL,
	OPEN_EXISTING,
	FILE_ATTRIBUTE_NORMAL,
	NULL);
	
	/* if there is a problem opening the file, a call to GetLastError will tell you what it is */
	
	To read from the file:

	`bTest= ReadFile(hFile, dwBuffer, sizeof(DWORD)*256, &dwNumRead,NULL);`
	`/* bTest will be TRUE if the read is successful.`
	`If false, take a look at GetLastError */`
	`[...]`

Source References	• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/fileio/base/createfile.asp[2] • Pazera, Ernest. File I/O in Visual C++[3] (1998).
Recommended Resource	
Discriminant Set	**Operating System**: • Windows **Languages**: • C • C++

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com